CLAIMS

What is claimed is:

1. A computer system, comprising at least one processor, a system memory coupled to said processor, and at least one input/output device coupled to said processor, wherein the computer system processor is configured to execute:

an operating system with at least two protection levels;

a watchdog driver;

at least one computer application; and

a reset service;

wherein the watchdog driver observes at least one application for a periodic message from the application and wherein if the periodic message is not received in a predetermined period of time, the watchdog driver instructs the reset service to initiate a reset procedure.

2. The computer system of claim 1 further executing:

a message passing interface configured to transmit signals between the two protection levels;

wherein the watchdog driver is configured to execute in one protection level and the application is configured to execute in another protection level and wherein the periodic message is transmitted from the application to the watchdog driver through the message passing interface.

3. The computer system of claim 2 wherein:

the message passing interface is a shared memory queue.

4. The computer system of claim 1 wherein:

the restart service is configured to close and restart the application upon receiving the instruction to initiate the restart procedure.

5. The computer system of claim 1 wherein:

the restart service is configured to restart the system upon receiving the instruction to initiate the restart procedure.

6. The computer system of claim 2 wherein:

the watchdog driver establishes timer events in the operating system scheduler that alert the watchdog driver when the predetermined period of time has expired.

7. An application watchdog, comprising a restart service operating in the user mode of a computer operating system and a watchdog driver operating in the kernel mode of the computer operating system comprising, the watchdog driver comprising:

a system thread configured to monitor a plurality of user applications operating in the user mode of the computer operating system;

a first IOCTL signal interface for communicating control signals between the watchdog driver and each of said user applications; and

a second IOCTL signal interface for communicating control signals between the watchdog driver and the restart service;

a communication interface for coordinating timer events with the operating system scheduler corresponding to each of said applications and indicating when each of said applications is presumed to be unresponsive;

wherein if the system thread does not receive a message from one of said applications within an allotted period of time, the timer event alerts the watchdog driver that the allotted time has elapsed and the watchdog driver signals the restart service to restart that application.

8. The application watchdog of claim 7 wherein:

if the system thread does receive a message from one of said applications, the timer event corresponding to said application is updated to reflect the current time plus the allotted period of time.

9. The application watchdog of claim 7 wherein:

the messages from said applications are sent periodically by the applications and directed specifically to the watchdog driver.

10. The application watchdog of claim 7 wherein:

the messages from said applications are sent to the watchdog driver via a message passing interface between the user mode and kernel mode.

11. The application watchdog of claim 7 wherein:

the restart service is further configured to execute the following;

user notification;

error logging; and

multiple application reset.

12. The application watchdog of claim 7 wherein:

the plurality of applications are prioritized by a computer user to permit varying levels of watchdog protection.

13. The application watchdog of claim 7 wherein:

the restart service is further configured to perform a system reset.

14. A method of detecting and restarting an unresponsive computer application, comprising:

executing the application in a first protective layer of a computer operating system;

executing an application watchdog driver in a second, more protected, protective layer of the computer operating system;

establishing a message passing interface between the application and the watchdog driver;

periodically transmitting signals from the application to the message passing interface;

executing a system thread in the watchdog driver that is configured to monitor the message passing interface for the periodic signals from said application or other applications; and

executing a reset service that is configured to terminate and restart one or more applications;

wherein if the system thread fails to detect the periodic signals from the application for a pre-configured amount of time, the watchdog driver initiates a command to the restart service to terminate and restart the application.

15. The method of claim 14 wherein:

the message passing interface is implemented as shared memory queues.

16. The method of claim 14 wherein the initialization of the watchdog driver comprises:

loading the watchdog driver as the operating system loads following a computer system boot;

loading and creating an initial input/output control signal interface that establishes the message passing interface; and

loading and creating a second input/output control signal interface for communication with the reset service.

17. The method of claim 16 wherein the initialization of the reset service comprises:

loading the reset service in the first protective layer of the computer operating system; and

calling the watchdog driver via the second input/output control signal interface to verify communication with the watchdog driver.

18. The method of claim 17 wherein the initialization of the computer application comprises:

linking the application with a dynamic link library;

calling the watchdog driver via the dynamic link library and through the initial input/output control signal interface to validate the message passing interface;

sending application location and identification information to the watchdog driver; and

forwarding the application location and identification information to the reset service.

19. The method of claim 14 further comprising:

setting up timer events with the operating system scheduler that alert the watchdog timer when the pre-configured amount of time has elapsed.

20. The method of claim 19 wherein:

when a periodic message is received by the system thread, resetting the timer events.

21. A computer system, comprising:

an operating system with at least two protection levels;

a kernel mode watchdog driver;

at least one user application; and

a user mode reset service;

wherein the watchdog driver monitors at least one application for a periodic message from the application and wherein if the periodic message is not received in a predetermined period of time, the watchdog driver instructs the reset service to initiate a reset procedure.

22. The computer system of claim 21 wherein:

the reset procedure comprises closing and restarting the application.

23. The computer system of claim 21 wherein:

the reset procedure comprises restarting the operating system.

24. The computer system of claim 21 wherein:

the watchdog driver creates timer events in the operating system scheduler that alert the watchdog driver when the predetermined period of time has expired.